MANSOURA UNIVERSITY | Jan. 2010 | 2nd. YEAR COMM. ENG.
FACULTY OF ENGINEERING | Prof. F.W.Zaki | LOGIC CIRCUITS (2)
DEPT. OF ELECT. & COMM. ENG. | TIME ALLOWED = 3 HRS

Answer ALL Questions (a total mark of 80 is equally distributed among all questions)

I-a) Construct a 5X32 decoder with four 3X8 decoders with Enable and one 2X4 decoder. Show your implementation in block diagram form with detailed connections.

b)Given four 32X8 ROM chips with Enable input, and each chip has its own address decoder. Show the external connections necessary to construct a 128X8 ROM with the four chips and one 2X4 decoder.

II-a)Design a nonbinary sequence counter that counts in the sequence 0, 1, 3, 7, 6, 4 and repeats using T-flip flops. Treat the unused states as don't care conditions. Analyze the final circuit to ensure that it is self correcting. If it is nonself correcting , modify the circuit to make it self correcting.

b)Show an implementation of a 4-bit binary sequence UP-DOWN synchronous counter using T-flip flops.

III-a)Show an implementation of a 4-bit BIDIRECTIONAL shift register with parallel load using D-type flip flops and four 4X1 multiplexers. Show also the connections for clear and clock inputs.

b)A sequential circuit consists of two J-K flip flops (A, B), one input X, and one output Y. Complete its state table shown, then derive the state equations, the state diagram, and the logic implementation of the circuit.

| Present state | | input | Next state | | output | f.f. inputs |
|---|---|---|---|---|---|---|
| A | B | X | A | B | Y | |
| 0 | 0 | 0 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 0 | 0 | 1 | |
| 0 | 1 | 0 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | 1 | |
| 1 | 0 | 1 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 1 | 0 | 1 | |

IV-a) With the aid of graphical illustrations show the data flow and instructions flow in a µP and explain the sequence of steps involved.

b) In a certain µP, the main program runs from address D110 to D193. Three nested subroutines have their start and end addresses as:

subroutine A→ A200 to A235
subroutine B→ A239 to A258
subroutine C→ C1A1 to C1A9

The BRANCH instructions are at the following addresses:

subroutine A→ D122
subroutine B→ A232
subroutine C→ A249

i-If the CPU internal stack registers are used, graphically show what the stack registers will contain after each push and pop operation.

P.T.O

**ii-If a memory type stack whose addresses run from B138 to B13D (6-deep stack) is used. Graphically show the contents of the stack, the stack pointer, and program counter after all branches and returns for all three subroutines.**

**V-a)In a certain μP the effective addresses of some locations in data memory are pre-stored in scratch pad locations as follows:**

| Memory address | Scratch pad address |
|---|---|
| 112B | 01 |
| 13AC | 02 |
| 146E | 03 |
| 003A | 04 |
| 003B | 05 |
| 003C | 06 |

**The data for operands A, B, and C are stored in the locations: A in location 112B, B in location 13AC, and C in location 146E. Write a program using the codes given in the tables attached to first clear the accumulator, then perform the following operations using the previously cleared accumulator, (indirect addressing mode is used):**

1. **A+B=S, and S to be stored in the memory location 003A**
2. **S - C=D, and D to be stored in the memory location 003B**
3. **D+S=Z, and Z to be stored in the memory location 003C**

**b)The accumulator of a certain μP contains $44_H$. The location to be accessed in the data memory contains $33_H$. The instruction which uses the program counter relative addressing mode, is fetched from address $02D7_H$ of the program memory. The A field of the instruction contains $7D_H$. The O field of the instruction commands the μP to transfer the contents of the accessed data memory location into the accumulator and add it to the previous contents. Determine the contents of instruction register, the program counter after the instruction is fetched, the MAR, and the accumulator after the instruction is executed.**

**GOOD LUCK**

Table 5-2 THE O-FIELD CODES FOR THE GEMINI-A INSTRUCTIONS

| *Binary* | *HEX* | *Instruction* |
|---|---|---|
| 0001 | 1 | Store, (ACC) → MEM |
| 0010 | 2 | Load, (MEM) → ACC and add<br>(MEM) + (ACC) → ACC |
| 0011 | 3 | Load, (MEM) → ACC and subtract<br>(ACC) − (MEM) → ACC |
| 0100 | 4 | Clear ACC |
| 0101 | 5 | Load index register (if the M code is the immediate address mode, the second word of that instruction will be loaded into the index register). |
| 0110 | 6 | Load scratch pad immediately. The address of the scratch-pad register will be given in the A field of the instruction, and the operand to be loaded will be in the second word of the instruction. |
| 0111 | 7 | JUMP, unconditional |

Table 5-3 THE M-FIELD CODES FOR THE GEMINI-A INSTRUCTIONS

| *Binary* | *HEX* | *Instructions* |
|---|---|---|
| 0000 | 0 | Direct addressing |
| 0001 | 1 | Indirect addressing through scratch pad |
| 0010 | 2 | Indirect addressing through base page (Page 0) |
| 0011 | 3 | Page relative addressing (current page) |
| 0100 | 4 | ← Base page relative addressing |
| 0101 | 5 | ← Program counter relative addressing |
| 0110 | 6 | ← Immediate addressing (2-word instruction) |
| 0111 | 7 | Direct index addressing (2-word instruction) |
| 1000 | 8 | Indirect indexed addressing |
| 1001 | 9 | Indexed indirect addressing |
| 1111 | F | Nonmemory reference instruction—A field of the instruction is ignored |